CDF/DOC/TRIGGER/CDFR/7064
July 11, 2004

`

Silicon Vertex Trigger Upgrade[1]

J. Adelman, I. Furic, Y.K. Kim, M. Shochet, U.K. Yang
University of Chicago

T. Liu
Fermilab

I. Roffilli, L. Sartori, F. Schifano, R. Tripiccione
Universita' di Ferrara

A. Cerri
Lawrence Berkeley National Laboratory

A. Bardi, F. Bedeschi, R. Carosi, S. Galeotti, P. Giannetti, M. Piendibene, L. Ristori, F. Spinella
INFN Pisa

G. Punzi
Scuola Normale Superiore di Pisa

A. Annovi, M. Bitossi, M. Dell'Orso, S. Donati, P. Giovacchini
Universita' di Pisa

M. Rescigno, L. Zanello
University of Rome

P. Catastini, A. Ciocci, S. Torre
Universita' di Siena

T. Maruyama
Univeristy of Tsukuba

J. Bellinger, D. Carlsmith, W. Chung, C.M. Ginsburg, R. Handler, G. Ott, L.G. Pondrom
University of Wisconsin

1. Introduction

The CDF Silicon Vertex Trigger[2] (SVT) adds important triggering capabilities to the CDF detector in Run II. It provides for the first time in a hadron collider detector the ability to trigger on *b* quarks based solely on the generic property that *b* hadrons are metastable, with lifetimes of ~ 1.5 psec. The input to the SVT is a list of drift chamber tracks found by the level-1 track trigger, XFT, and the axial hits from the silicon vertex detector (SVX). SVT track candidates contain an XFT track and hits on at least 4 SVX layers. After fitting, the impact parameter of the track relative to the accelerator beamline has a resolution comparable to the width of the beam (see figure 1). This provides good discrimination between tracks originating in the primary $p\bar{p}$ collision and those arising from the decay of *b* hadrons typically a few millimeters from the interaction point.

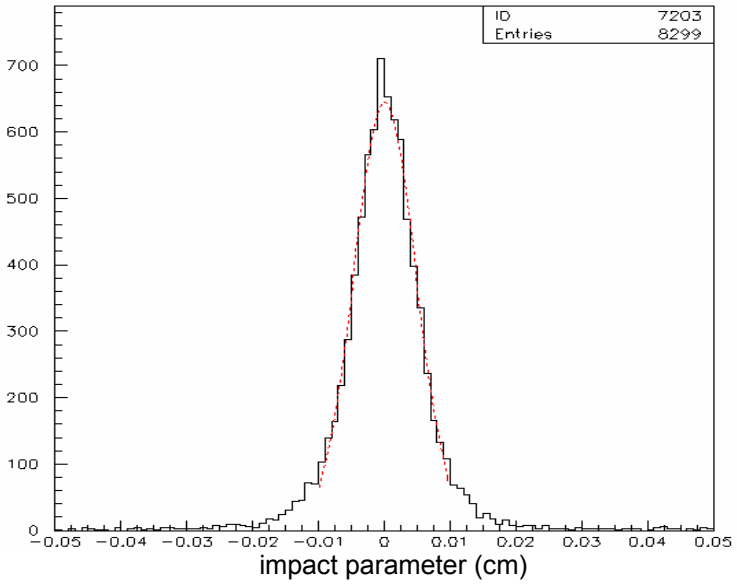

Figure 1: The impact parameter of primary tracks from $p\bar{p}$ collisions relative to the proton beamline as measured by the SVT. The σ of the distribution, ~50 μm, results from the 30 μm beam size and the 40 μm SVT resolution.

The SVT is important for both the high-P_T and *B* physics programs in CDF. Among the high-P_T triggers that require large impact parameter SVT tracks are three triggers that feed broad new physics searches: single high-P_T *b* jet, missing E_T plus *b* jet, and photon plus *b* jet. There is also a Higgs multijet trigger aimed at $H \to b\bar{b}$ decays, and the $Z \to b\bar{b}$ trigger which is important for setting the *b*-jet energy scale for both the top quark and Higgs masses. The CDF *B* physics program relies heavily on the SVT. Both multitrack and lepton plus track triggers, which are necessary for measuring B_s mixing, require large impact-parameter tracks. The first CDF Run-II PRL paper, reporting the world's best measurement of the D_s–D^+ mass difference, used data taken with an SVT trigger (see figure 2).

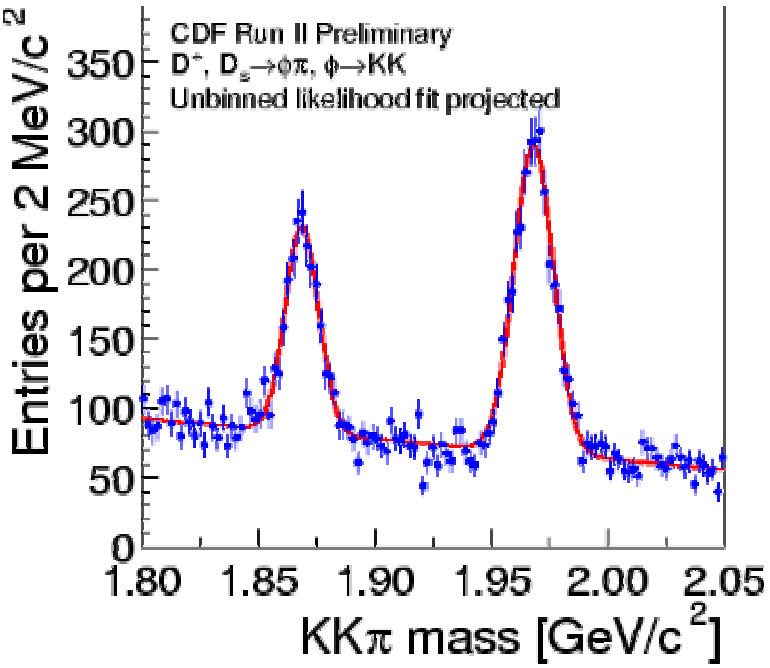

Figure 2: The $KK\pi$ invariant mass distribution used to determine the mass difference between the D_s and D^+ mesons. The data come from a trigger that requires two SVT tracks with large impact parameter.

The expected increase in luminosity in Run IIb requires significant upgrades to many CDF trigger and data acquisition systems. This is even more critical now because the impact of multiple interactions will be much greater than initially anticipated as a result of the decision to retain 396 nsec bunch separation rather than reducing it to 132 nsec. The data throughput from the detector to the level-3 trigger processing farm and from there to the Feynman Center is being significantly increased. The upgrade of the level-2 trigger using the Pulsar boards[3] will reduce the time to combine the information from the various level-2 data sources and make the final trigger decision. The limiting factor in Run IIb triggering will be the execution time of the SVT. The goal of the SVT upgrade is to reduce that time and thereby increase the physics capability of CDF.

2. The Physics Case

In the trigger, the critical parameter is deadtime. At level 2, the deadtime depends on the level-1 trigger rate multiplied by the time it takes to make a decision at level 2. At current luminosities (8×10^{31} cm^{-2} sec^{-1}), both high-P_T and B physics level-1 triggers have to be prescaled because of SVT execution time. In this section, we show the severity of the problem at the design Run-IIb luminosity of 3×10^{32} and how the proposed SVT upgrade would mitigate the problem.

The SVT is divided into 12 azimuthal wedges that are processed in parallel. There are effectively two major SVT stages. The first reads in the silicon data and does track pattern recognition. To a good approximation, the time for this stage is proportional to the number of silicon hits in the busiest wedge. The second stage fits the track

candidates, with an execution time proportional to the number of hit combinations to be fit. Data taken with the SVT at different luminosities were used to determine the coefficients for the linear relation $t_{SVT} = aN_{hits} + bN_{comb}$. Figure 3 compares the prediction of this model with actual timing measurements. The agreement is good, allowing us to predict the performance of the existing SVT and the proposed upgrade at higher luminosities.

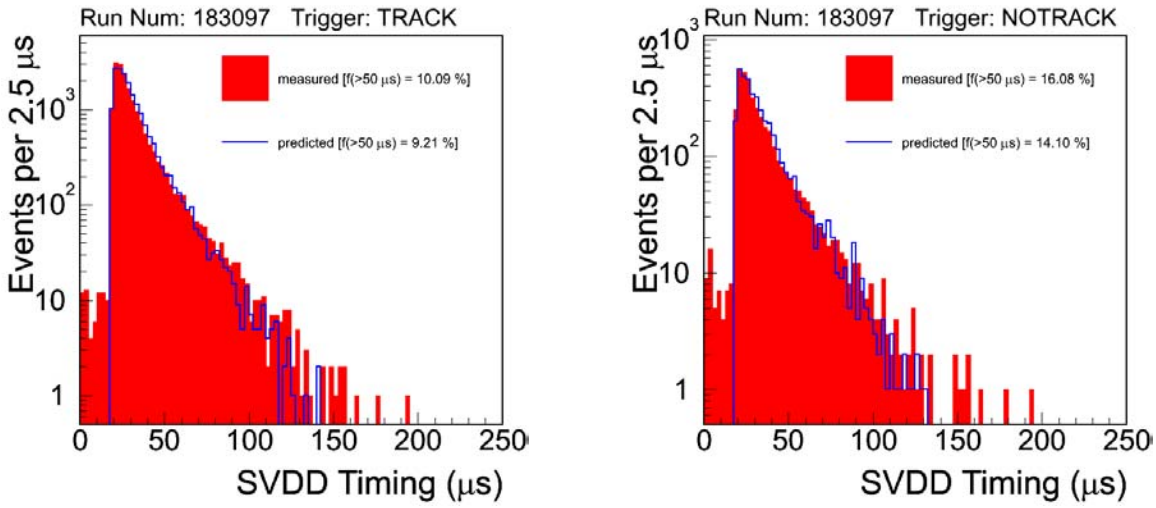

Figure 3: SVT execution time distribution in data (solid histogram) compared to the prediction from the linear model (curve). On the left is a mix of triggers that requires tracks in the level-1 trigger. On the right is a mix that does not require level-1 tracks.

In order to use this model to predict performance at high luminosity, we need the distribution of the number of SVX hits and track fit combinations as a function of luminosity. Data were taken at three luminosities between 1×10^{31} and 5×10^{31} with a special set of triggers ($Z \to b\bar{b}$ for high P_T and the two track *B* trigger). The hit and combination distributions were then extrapolated to 3×10^{32} and used as input in a queuing program that has been used in the past to understand the rate limitations of the multi-level CDF trigger system. It uses timing information for each stage of the trigger and includes the effect of the limited number of level-2 buffers.

Figure 4 shows three simulations compared to data taken at a luminosity of 5×10^{31} with a 2-track level-1 trigger using the SVT as it existed in May. The first simulation includes the Road Warrior, which removes duplicate tracks prior to fitting and has recently been implemented with Pulsar boards. The second is with the same system but at a luminosity of 3×10^{32}. The third is with the fully upgraded SVT at high luminosity. The resulting deadtime vs. level-1 rate is shown in figure 5. The maximum level-1 rates for 5% deadtime are tabulated in Table 1, showing that CDF will be able to increase its level-1 trigger rate by almost a factor of 2 after the SVT is upgraded. The validity of the results was tested at current luminosities. The rates in the table have uncertainties of approximately $\pm 30\%$ due to the extrapolation of a factor of 6 in luminosity.

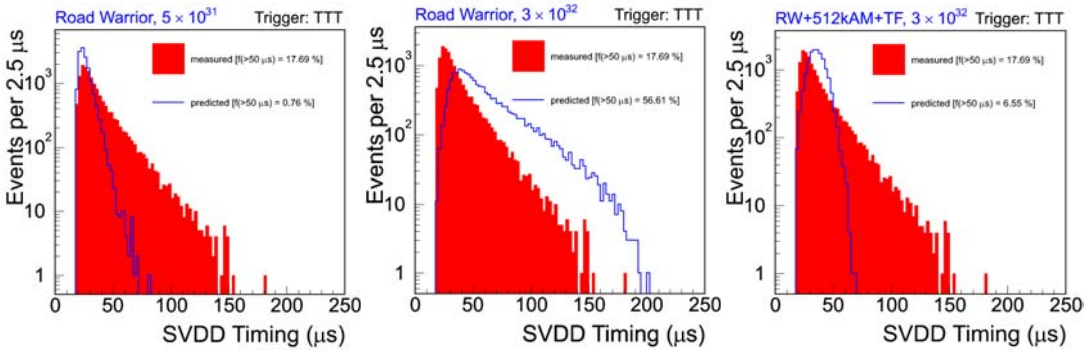

Figure 4: Comparison of SVT timing simulations with data taken with the existing baseline SVT at a luminosity of 5×10^{31}. The simulation on the left is at the same luminosity but with the recently added Pulsar Road Warrior included. The central figure is with this system but at a luminosity of 3×10^{32}. The right figure shows the effect of the proposed SVT upgrade at high luminosity.

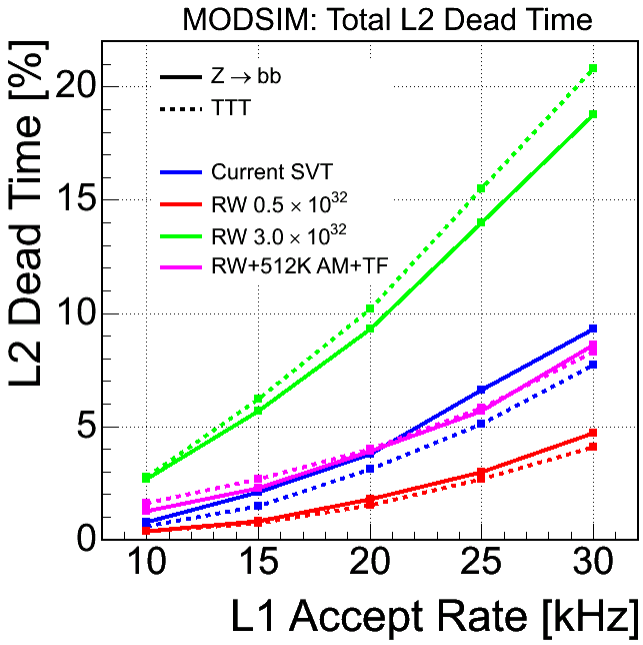

Figure 5: The level-2 deadtime as a function of the level-1 trigger rate for both a high-P_T trigger ($Z \to b\bar{b}$) and a B physics trigger (two track trigger) for four sets of conditions: in blue, the baseline existing SVT at a luminosity of 5×10^{31}; in red, the same with the addition of the Road Warrior; in green, the same as red except at a luminosity of 3×10^{32}; in magenta, the green conditions but with the upgraded SVT.

Current SVT	Upgraded SVT
13 KHz	23 KHz

Table 1: The maximum level-1 trigger rate for 3×10^{32} luminosity so that the level-2 deadtime does not exceed 5%.

In order to interpret these results, we need the level-1 trigger rates at the Run IIb design luminosity. Again we used data taken at current luminosities to extrapolate up to 3×10^{32}. These estimates may in fact be low because we fit to a constant trigger cross section plus a linear growth term. We assumed no quadratic term, even though with these multi-object triggers such terms are very possible. The expected level-1 trigger rates are shown in Table 2. The uncertainty in these rates is a factor of 50% because of the large luminosity extrapolation.

Trigger	level-1 trigger rate
$Z \to b\bar{b}$	26 KHz
Hadronic B decay	177 KHz

Table 2: The expected level-1 trigger rates at 3×10^{32} from one high-P_T trigger and one B physics trigger.

The comparison of Tables 1 and 2 makes it clear that at the design Run IIb luminosity CDF will be deadtime limited. Consequently the SVT upgrade will increase the amount of SVT triggered data CDF can collect by almost a factor of 2.

3. Overview of the existing SVT

Before presenting the proposed upgrade to the SVT, we review the existing device. A sketch for one of the 12 azimuthal wedges is shown in figure 6. Raw silicon hits are transmitted on optical fibers to Hit Finder boards that find hit clusters in each silicon layer and calculate their centroids. The Merger board then combines the silicon data from 3 Hit Finders with the drift chamber tracks in that wedge. The combined information is sent to both the Associative Memory system, which does the pattern recognition, and a Hit Buffer which stores the hits until a pattern is found with sufficient hits to be deemed a track candidate. There are three pattern recognition boards, a Sequencer that controls the operation and two Associative Memory (AM) boards that find track candidates. The Sequencer converts silicon hit coordinates into coarser resolution superstrips, typically 500 μm wide, a resolution appropriate for pattern recognition. The AM boards contain AM chips which are content addressable memory. There are 32K patterns stored, each containing the superstrip (SS) number for a drift chamber track and for each silicon layer. As each hit passes through the AM boards, all patterns see it at the same time. If the SS number is the same as the one stored in the pattern for that layer, then the layer is marked as on. When all silicon and drift chamber data have passed through the system, each of the 32K patterns has a word containing the layers that were

hit. Majority logic selects those patterns ("roads" in the figure) with a drift chamber track and the requisite number of silicon hits and passes them to the Hit Buffer board.

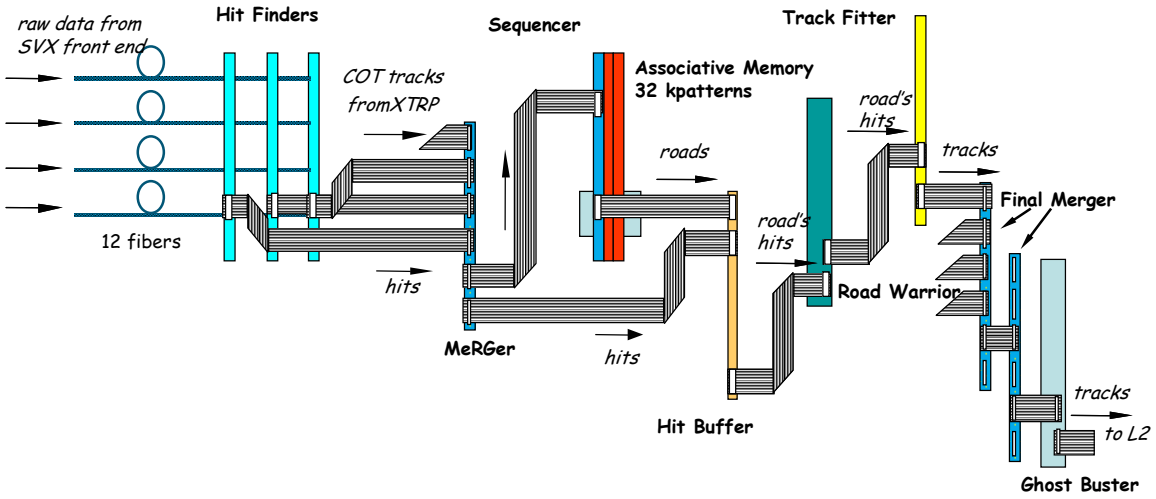

Figure 6: The SVT boards for one of 12 azimuthal wedges.

As each road enters the Hit Buffer, the silicon hits are retrieved at full resolution and passed to the Track Fitter along with the road number. Because the roads are quite narrow, each one corresponds to an approximate track momentum, azimuth, and impact parameter. This permits track fitting to be done with the requisite resolution using a linear approximation. The 6 inputs (4 silicon hits plus the drift chamber track azimuth and curvature) are used to produce 6 outputs (impact parameter, improved curvature and azimuth, plus 3 constraints that are combined into a fit χ^2). Tracks that pass a χ^2 cut are sent to the Ghostbuster board. If there is more than one silicon track associated with a single drift chamber track, the Ghostbuster selects the one with the smallest χ^2. Also shown in the figure is the recently added Road Warrior, a Pulsar board that removes duplicate track candidates prior to fitting.

4. The SVT Upgrade

As luminosity rises, multiple interactions increase the number of hits in the SVX. Thus the time to process all of the silicon hits increases (time/hit = ~35 nsec). However the more serious problem is the increase in the number of track candidates that have to be fit (time/fit = ~300 nsec). In each road, the number of fits that has to be done is the product of the number of hits in each silicon layer. As the hit density in the silicon increases, the number of fits can become quite large.

The SVT upgrade aims to both reduce the number of fits and to perform each fit more quickly. The former is done by reducing the width of roads by increasing the total number of roads per wedge from 32K to 512K. The latter is achieved by increasing the speed of the track fitting board. Table 1 shows the improved SVT performance with these upgrades. The increase in the amount of data CDF can record is very significant.

The existing muon trigger requires a high quality track to match a stub in the muon chambers. At present the XFT can find tracks that exit the COT before reaching

the outer superlayer. However it is expected that in Run IIb, the XFT will require hits in the outer superlayer. This will reduce the efficiency for muons in the rapidity range between 1.0 and 1.5 by ~50%. Using silicon tracks instead of COT tracks for muon matching could restore the performance of the muon system.

There is an additional advantage to upgrading the SVT that is more difficult to quantify. The upgrade provides insurance against further degradation of our central drift chamber (COT). Without all of the COT superlayers, the XFT has a larger fake rate that increases the burden on the SVT. Such XFT tracks also have worse pointing resolution into the silicon, implying a larger number of patterns for the same overall track efficiency.

The SVT upgrade is designed to require a minimum of new hardware. The increase in the number of roads is achieved with a new AM system, the AM++. The new AM++ boards themselves have been under design for a while based on work that was done for the LHC. In fact new prototype AM chips will be produced shortly. The new Sequencer will use a Pulsar board, which was designed for and is now being used in the CDF level-2 upgrade. The Pulsar design is very flexible, with mezzanine cards for various input and output protocols and three powerful FPGAs for data manipulation. A simple mezzanine card containing memory chips and firmware for the Pulsar FPGAs are the major needs for the AM Sequencer. The AM Sequencer will also carry out the Road Warrior function, to remove redundant track candidates prior to track fitting.

The existing Hit Buffer cannot be used with the proposed increase in the number of track patterns. The Hit Buffer functionality will be transferred to a Pulsar board. The additional needed memory will be located on a mezzanine card.

The new Track Fitter will also use the Pulsar board, taking advantage of its factor of 3 higher clock speed compared to the current Track Fitter. A mezzanine card similar to those used on the other SVT Pulsar boards will be utilized here. The FPGA firmware will be based on the firmware in the current Track Fitter.

The block diagram of the upgraded SVT with the 3 Pulsar boards per wedge is shown in figure 7.

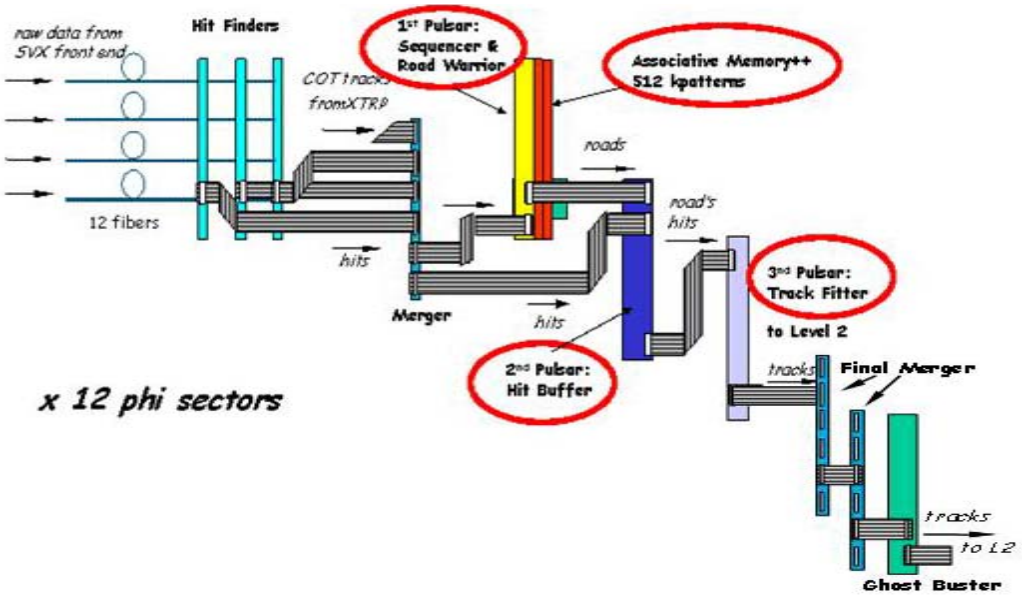

Figure 7: Block diagram of the upgraded SVT.

In addition to the hardware tasks noted above, we will have to update the online, monitoring, and simulation software to reflect the properties of the new boards.

Commissioning the new system will proceed in three steps. New boards will first be tested in the Pulsar test stand. Since Pulsar boards are universal transmitters and receivers, a new board can be fed simulated data from one Pulsar and drive its output into another Pulsar. This allows full functionality, full speed testing before a board is brought into the trigger room. When this is complete, the new board can be tested with the rest of CDF without interrupting data taking. An existing SVT splitter board can parasitically feed the real signals into the new board. Its output can be compared to expectation. Finally when these tests are complete, the new boards can replace the old ones in the operating system.

Phased installation is also possible. When the Track Fitters are ready, they can be installed, even if the new AM++ system is not ready. With the new Track Fitters in place, the SVT will operate faster than at present. The AM++ installation requires the new Hit Buffers and Track Fitters because of the increase in the number of roads.

4.1 AM++ board

We have designed a new Associative Memory Board (AM++) for track finding. It works at a clock frequency of 40 MHz and has a modular structure, consisting of 4 smaller boards, the Local Associative Memory Banks (LAMBs). Each LAMB contains 32 Associative Memory chips, 16 on each side of the board. The AM++ board is sketched in figure 8. The structure of the LAMB is also shown. The AM chips come in PQ208 packages and contain the stored patterns and read-out logic.

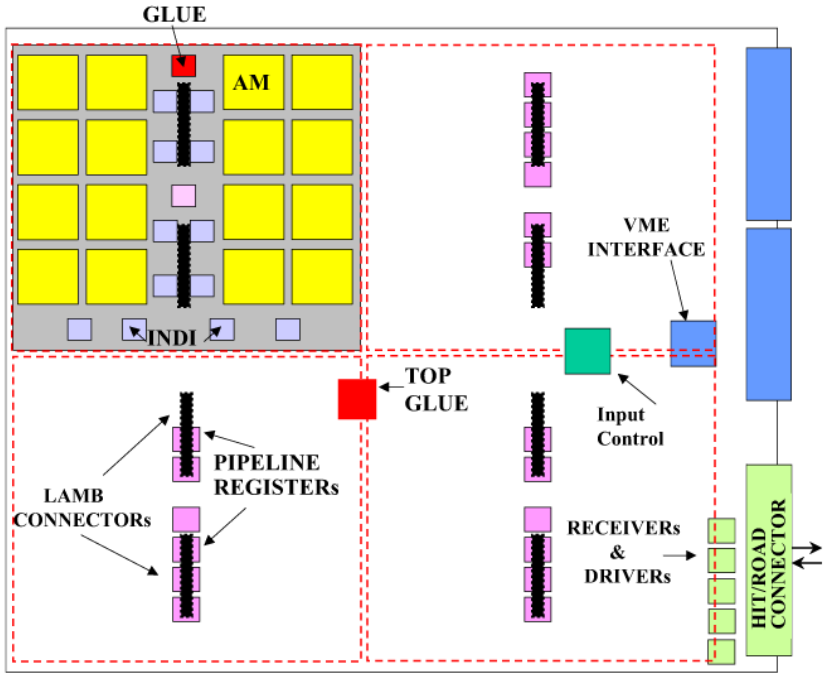

Figure 8: AM++ board layout. Dashed lines show areas occupied by LAMBs. On the top left corner, the colored area shows details of one LAMB.

It is important to remember that in order to contain the pattern bank within an acceptable size the AM operates at a coarser resolution than the actual detector. This is usually done by clustering single contiguous detector channels into larger superstrips. In the following, we will call "hits" the addresses of fired superstrips in each layer, and "roads" the coarse resolution trajectories (each road corresponds to an array of superstrip addresses – one per detector layer).

When an AM++ board starts to process an event, the hits are received by the input control chip, and then they are simultaneously sent to the four LAMBs. As soon as hits are downloaded into the LAMBs, the locally found roads set the request to be read out (Data Ready). Once the event is completely read out, the LAMBs make the last matched roads available within a few clock cycles.

4.1.1 LAMB

Six hit buses, one for each detector layer, are fed in parallel to the four LAMBs and distributed to the 32 AM chips on the LAMB through 12 fan-out chips called Input Distributors (INDI). For the output road address bus, the AM chips on each LAMB are divided into 4 pipelines, each corresponding to a column in figure 9, consisting of 8 chips, 4 on the front and 4 on the back. Each AM chip receives an input bus where road addresses found in previous chips are propagated, multiplexes it with the road addresses internally found in that chip, and sends the output to the next chip. Signals propagate from one chip to another on the top and bottom PCB planes through very short pin-to-pin lines, without any vias.

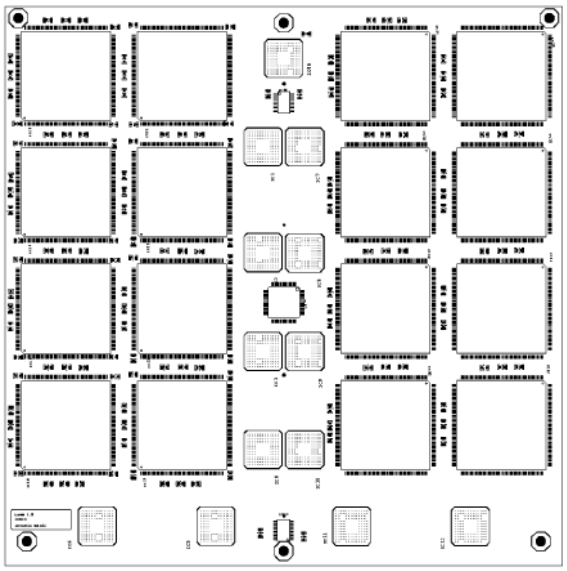

Figure 9: Top view of LAMB board.

The outputs of the 4 pipelines are then multiplexed into a single bus by a specially designed GLUE chip on each LAMB.

The LAMB board has been designed, optimized, simulated, placed, and routed with Cadence software. It represents a significant technological challenge due to the high density of chips allocated on both sides of the board and the use of the advanced Chip-scale packages (CSP) for the 12 INDI chips and the GLUE chip (see figure 9).

The CSP features a ball-grid array with a 0.8 mm pitch, connecting to a silicon die that is only 20% smaller than the package size. Such small pitch and large board density push the PCB geometry to the edge of available technology, with 20 mil blind vias, lines with minimum width and spacing of 5-6 mils in a 1.6 mm thick board with 8 routing layers. Figure 10 shows a portion of the LAMB PCB top layer. Next to the larger AM chip on the left, 4 INDI chips are shown. Note the small dimensions compared to the inch ruler. Successful operation has been tested at a clock frequency of 40 MHz using FPGA associative memories (PAM, programmable AM), which are pin compatible with the future standard cell chip. Patterns are downloaded through the VME controlled JTAG port. Chains of 4 AMs each are downloaded in parallel to reduce programming time. The VME 32-bit wide data transfer allows us to program 32 chains in parallel for a total of 4 LAMBs. Downloading time is a few seconds.

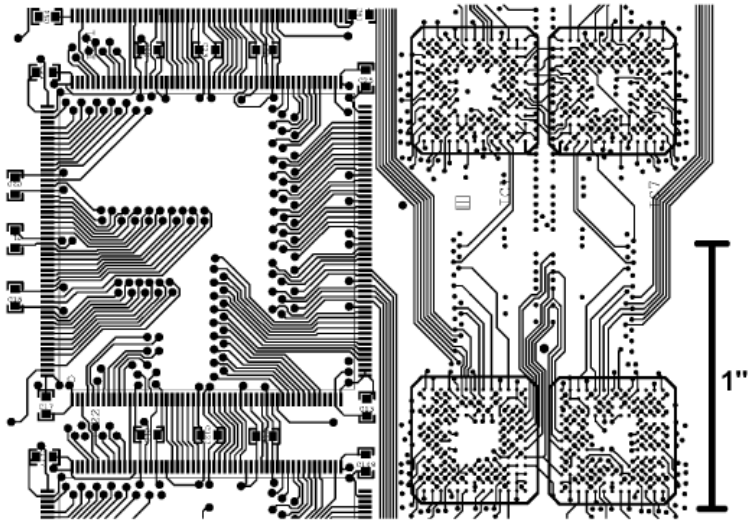

Figure 10: A closer view of the LAMB PCB.

4.1.2 AM chip

We are producing the upgraded associative memories using standard cell technology to achieve the maximum pattern density without resorting to full custom design. We are using the UMC 0.18 micron technology since it is the most convenient. A 10x10 mm die contains 5000 “CDF patterns” (patterns for a six layer detector).

We are starting with a Multi Project Chip (MPC) to get a low cost prototype. For the small number of required chips (~2000), a small volume production is convenient. The small volume production uses the same MPC masks used for the prototype (once the prototype is validated). Although this procedure wastes silicon area belonging to other projects in the Mult Project Wafer (MPW) run, it would save the cost of expensive new

masks necessary for a full pilot run ($250K). In this case (MPW) each wafer produces at least 30 chips and costs approximately $3300 for IMEC [4], which is going to produce our prototypes. A production of 1500-2000 chips will require two MPW runs of 25 wafers each, at a cost of $165K (IMEC).

The standard cell chip has been built pin compatible with an FPGA chip, so that extensive board tests can be prepared before receiving the standard cell prototype. For the implementation of the AM chip, we have chosen a commercial low cost FPGA family (Xilinx Spartan 0.35 micron process). The FPGA approach allows a larger degree of flexibility in the prototyping and testing phase of the project. Details of the Spartan family can be found on the Xilinx web page, http://www.xilinx.com.

The FPGA AM chips have been logically designed with the same VHDL code that defines the standard cell chip and have been mapped into FPGAs with Synopsis. The pattern density has been drastically reduced to 2 patterns to obtain a logically equivalent chip working at high frequency (40 MHz).

4.2 AM Sequencer and Road Warrior board

The AMS-RW card will implement and enhance the functions of two boards installed in the current SVT: the Associative Memory Sequencer (AMS) and the Road Warrior (RW). The AMS will be the operating sequencer for the AM++ associative memory boards, and the RW function will eliminate redundant track candidates before track fitting. The greatly increased number of track patterns, the protocol for the new AM chips, and the necessity of increasing processing speed require a new AMS-RW board.

The plan is to use a Pulsar board, which is highly reconfigurable and complies with CDF and SVT standards, to implement both the AMS and RW functions. A custom mezzanine card to be plugged into the Pulsar will satisfy the memory needs of the AMS-RW algorithms. The firmware for all the necessary logic will be contained in the 3 large FPGAs on the Pulsar.

4.2.1 Data flow

The AMS-RW communicates with up to two AM++ boards using a dedicated bus (AM bus) on the P3 backplane connector. The data flow is handled using a section of the firmware, GLUE, which completely hides the AM board details. The AMS-RW allows the AM++ board to communicate with the rest of SVT via dedicated connectors on the front panel following SVT communications protocol. Data flows in and out of the AMS-RW in data streams, one for input (Hits) and one for output (Roads).

In the input stream each packet is called a “hit”. Each word contains a hit coordinate (18 bits) and a layer number (3 bits). Hits may be one word long or more depending on what kind of detector the hit is coming from (SVX or XFT). Under normal circumstances (see Test Mode description below) data flow from the Hits stream to the AM bus and from the AM bus to the Roads stream. Incoming hits are sorted into a number of classes according to coordinate value ranges. These classes are called superstrips. The mapping between each hit and the corresponding superstrip is obtained through a 128k x 16 bit lookup table. Superstrips are sent by the AMS to the AM++

board which compares them on the fly with the AM content and keeps track of matches. Whenever a stored pattern is matched in the requisite number of layers, this is signaled to the AMS, which in turn instructs the AM++ board to send out the address of the pattern and also the list of the matched superstrips inside the pattern (bitmap). Each of these addresses is complemented with a bit field that identifies the sending AM++ board to form a "road" which is sent to the Road Warrior section.

The AMS can request the Associative Memory board to output only roads in which all 5 silicon layers are hit (5/5 mode) or allow roads with a missing layer (4/5). For the latter mode, the tracking efficiency is larger but SVT processing time increases. A large part of this increase in timing is due to "ghost" roads. When a track produces a hit on each of the five silicon layers, there are many output roads, one containing all five superstrips and roads containing different combinations of 4 hit superstrips. Without the Road Warrior function, duplicate tracks are eliminated after the Track Fitter. With the Road Warrior, the duplicate tracks are eliminated before the Track Fitter, significantly reducing fitting time.

The Road Warrior algorithm employs associative memory, widely used throughout SVT. It stores the SS combinations (patterns) of found roads in a small associative memory built on the fly when roads are sent to the output. Empty SS's are identified and flagged. Each road is allowed to proceed to the output only if it differs from all roads previously sent in at least one non-empty SS. Whenever the road number is received, it is used as an address to a lookup memory, the Associative Memory Map (AMMAP), which stores for each layer the superstrip associated with the road. The AMMAP is implemented as 1M x 36 bit RAM. Each pattern has associated with it 6 superstrips (one for each layer), each one 12 bits wide, so each pattern needs 2 locations in the AMMAP. The RW logic includes a 72 bit wide associative memory (AMRW), 64 locations deep, which easily fits in one of the Pulsar card FPGAs. The output of the AMMAP, after receiving a road and applying the bitmap, is compared to all the stored roads in the AMRW. If there is a match, the newly received road is a duplicate and is discarded. If there is no match, the AMMAP output is copied into the next available location in the AMRW, and the road is sent to the output engine. At the end of the event, the AMRW is cleared.

The output engine also calculates the parity of the output stream and sends the end-of-event word. The dataflow in the AMS-RW will be implemented as a multi-step pipeline, supervised by a finite state machine. Figure 11 shows the block diagram of the AMS-RW.

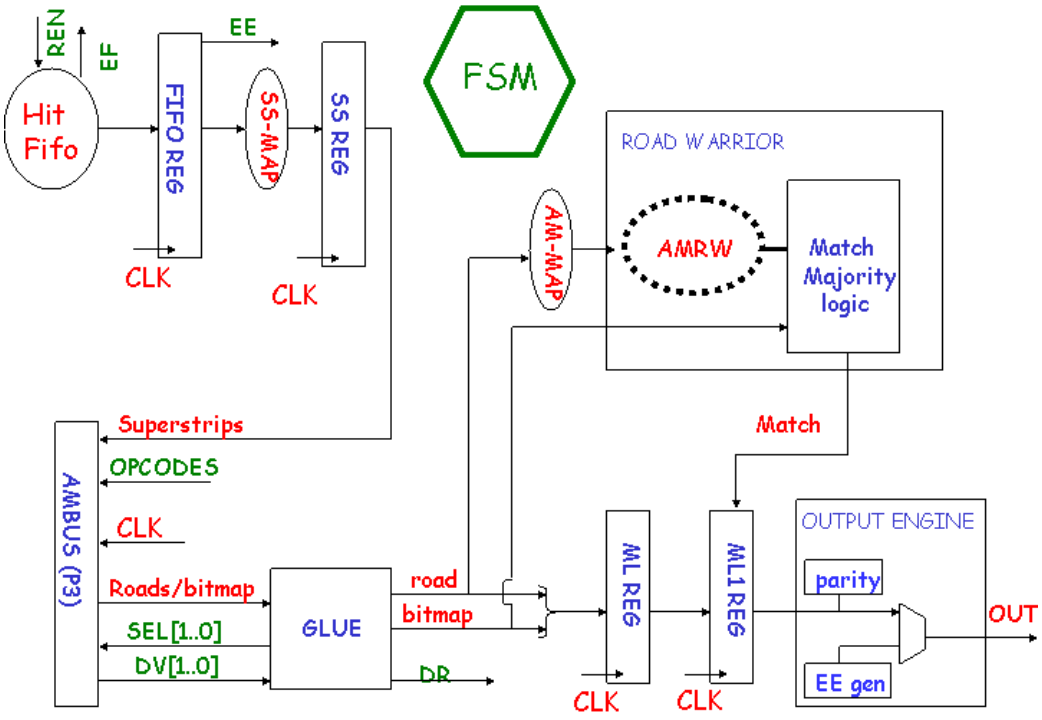

Figure 11: AMS-RW architecture

4.2.2 Hardware implementation

The AMS-RW card is accessed through a slave VME interface, already present in the pulsar board. The firmware implements a "test mode" where the data flow is stopped. The lookup tables can be written through VME only if the card is set in test mode. All the standard error handling features of SVT are implemented in the AMS-RW. The standard SVT spy buffer memories, which spy on the input and output streams, are implemented in the SRAM chips on the Pulsar.

No hardware modifications are needed on the Pulsar board, so that spares can be in common with the other Pulsar boards in the system. The needed mezzanine board is quite simple and will connect to the central mezzanine connectors (connectors 2 and 3). It will contain the SSMAP and AMMAP static RAM chips. Figure 12 shows the implementation of the AMS-RW inside a Pulsar board.

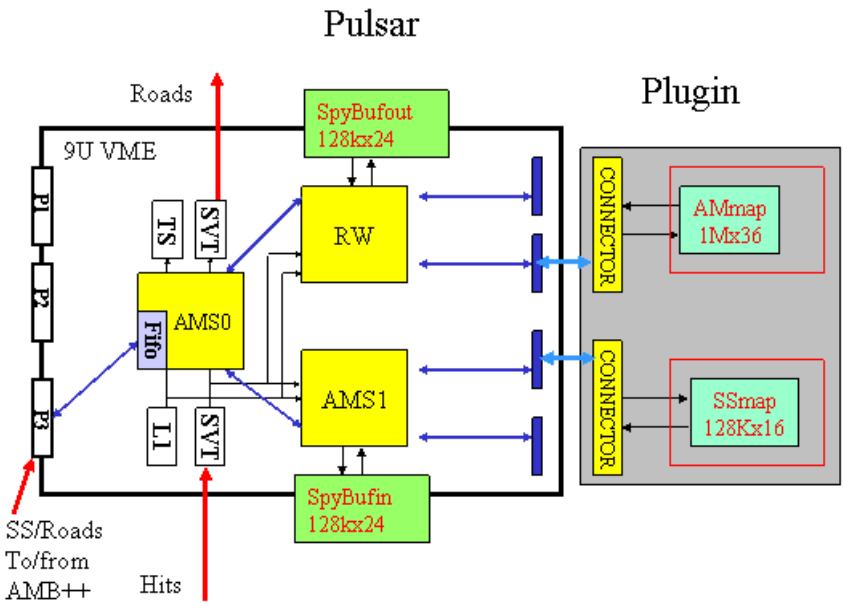

Figure 12: AMS-RW on the Pulsar board.

4.3 Hit Buffer

Within the Hit Buffer, hits are stored in the Hit List Memory (HLM). A structured data base is built on the fly so that each road number can then be used as a key to directly access lists of hits. Incoming hits are sorted into superstrips, with one hit list maintained for each superstrip. All of these lists are built on the fly in the Hit Buffer as the hits are received on input.

During the "Write Mode", hits are copied from the input and immediately organized in the Hit List Memory, depending on the superstrips they belong to. Roads are made of superstrips (one superstrip per layer). During the "Read Mode", each road received by the Hit Buffer requires that the Hit Lists corresponding to that road are sent from the Hit List Memory to the output stream, one superstrip (*i.e.* one layer) after the other.

To provide a very fast access to the Hit List Memory, the Hit Buffer has large look-up tables where pointers into the Hit List Memory are stored. The "SuperStrip Map" (SS Map) provides the starting address of the Hit List where each hit must be written when it arrives (Write Mode), while the "Associative Memory Map" (AM Map) provides the starting addresses of the Hit Lists to be sent on output for each road (Read Mode). The contents of the SS Map and of the AM Map are essentially the same, but the two maps implement different data base structures with different addressing schemes.

These look-up tables work in the following way:

- **SS Map: from hits to superstrips**

  The superstrip each hit belongs to is determined by the layer number and the value of the hit coordinate. The mapping from coordinate to superstrip is defined by a look-up table, the SS Map, implemented as a RAM where the 17 most

significant bits of the coordinate are used as an address and the corresponding superstrip ID is stored as the contents. Two pieces of information are encoded in the superstrip ID:

- the starting address of the relevant Hit List in the Hit List Memory (base address),
- the space reserved for the Hit List of that superstrip in the Hit List Memory (size).

Referring to the former, we say that the superstrip ID points to a memory location, as shown in Table 3. A word counter exists for each superstrip to record the current length of the corresponding Hit List. All of these counts are stored in a separate Hit Count block.

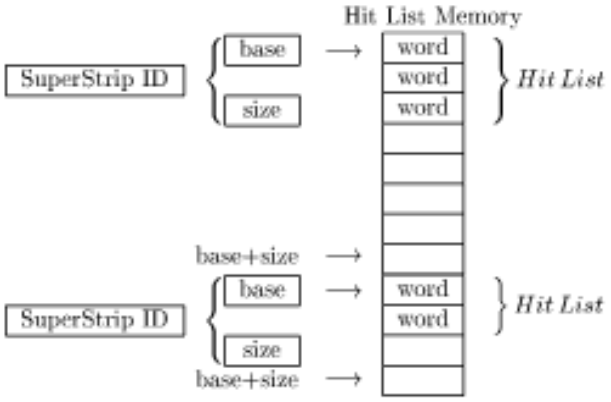

Table 3: Superstrip ID and Hit List structure in the Hit List Memory. Two pieces of information are encoded in the superstrip ID: base and size. Base is the starting address of the memory space reserved for the superstrip. Size is the number of reserved memory cells.

- **AM Map: from roads to superstrips**

Each road defines an array of superstrips, one superstrip per detector layer. We say that a road is made of a number of superstrips. The number of layers is determined by the physical structure of the detector, and the Hit Buffer is capable of accommodating from one to seven layers. Roads are identified by integer numbers. The correspondence between road number and superstrips is defined by a look-up table, the AM Map, implemented as a RAM where the road number and layer number are used as an address and the corresponding superstrip ID is stored in each location.

When a road number is received, it is used to access the look-up table and sequentially retrieve the pointers to all Hit Lists corresponding to the relevant superstrips, one per detector layer. These pointers are used to access the corresponding lists. All hits retrieved in this way are queued to form an output packet. The last word of the packet contains the road number

4.3.1 Hit Buffer Architecture

The hit buffer architecture is sketched in figure 13 and listed below.

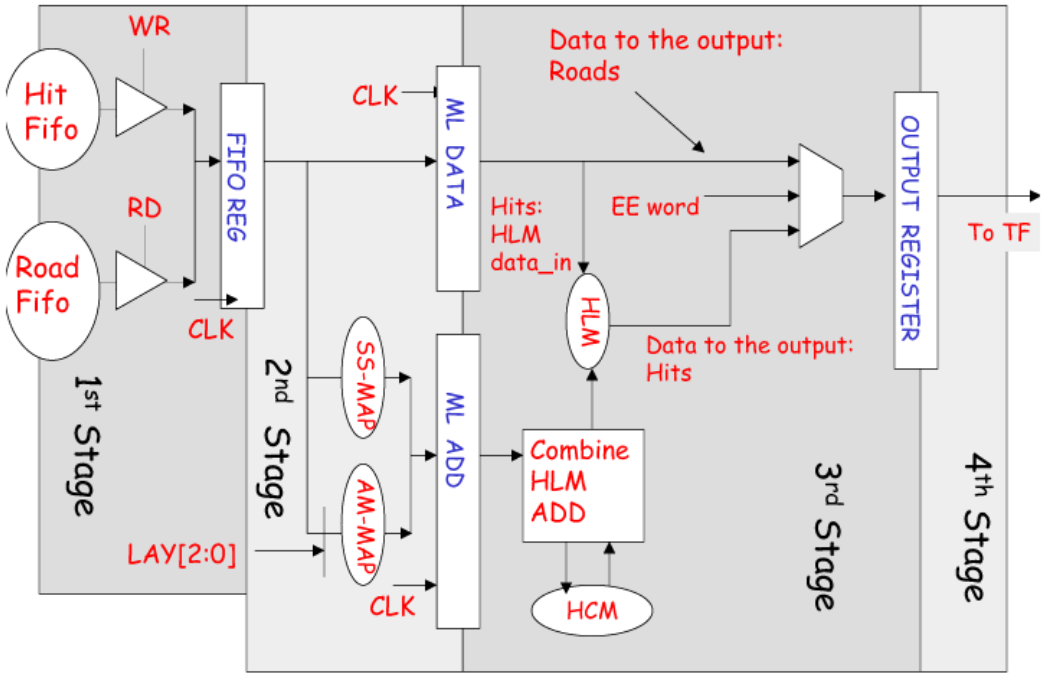

Figure 13: Hit Buffer architecture.

- The Hit Buffer is synchronized by an internal clock signal, with all state transitions occurring on the clock rising edge.
- Input data are asynchronously loaded by upstream modules into two FIFO buffers, the Hit-FIFO and the Road-FIFO. These FIFO's are read via two FIFO Controllers that convert the asynchronous FIFO protocol to the Hit Buffer internal synchronous pipeline protocol. This protocol and the FIFO Controller are described in detail in another document (SVT note 34).
- After initialization, the Hit Buffer is in Write Mode; it turns into Read Mode after EndEvent is received on the hit input stream. Write Mode is distinguished by WR flag = 1, Read Mode by WR flag = 0. During normal operation the Hit Buffer starts each event in Write Mode, switches after a while to Read Mode and stays there until the end of the event on the road stream, then switches back to Write Mode again.
- Data flow from the two FIFO's to the output connector is coordinated by a finite state machine (FSM). The FSM requests data from the input FIFO's and then latches them in an internal latch, called Middle Latch or ML in the following, which separates the look-up tables (SS Map and AM Map) from the Hit List Memory. Middle Latch is subdivided in two sections, ML DATA and ML ADD. When in Write Mode, ML DATA contains the hit to be written in the Hit List

Memory and ML ADD contains the superstrip ID with the pointer to the Hit List where it will be written (this information is fetched from the SS Map). When in Read Mode, ML DATA contains the road number and ML ADD contains the superstrip ID with the pointer (this information is fetched from the AM Map) to the Hit List to be fetched and sent on the output stream.

The Hit Buffer can therefore be thought of as a 4 stage pipeline (see figure 13):

1. from input FIFO's to FIFO Register under control of the FIFO controllers HFCTR and RFCTR (see SVT note 34)
2. from FIFO Register to Middle Latch, via SS Map and AM Map, under control of FSM
3. from Middle Latch to OUT Register, via the Hit List Memory, under control of FSM. Addressing of the Hit List Memory is done with the help of the Hit Count and of the Hit List Control logic (HLC).
4. from OUT Register to the input FIFO of the next board downstream, under control of FSM.

4.3.2 The Hit Buffer inside the Pulsar

Figure 14 shows the implementation of the HB inside the Pulsar board.

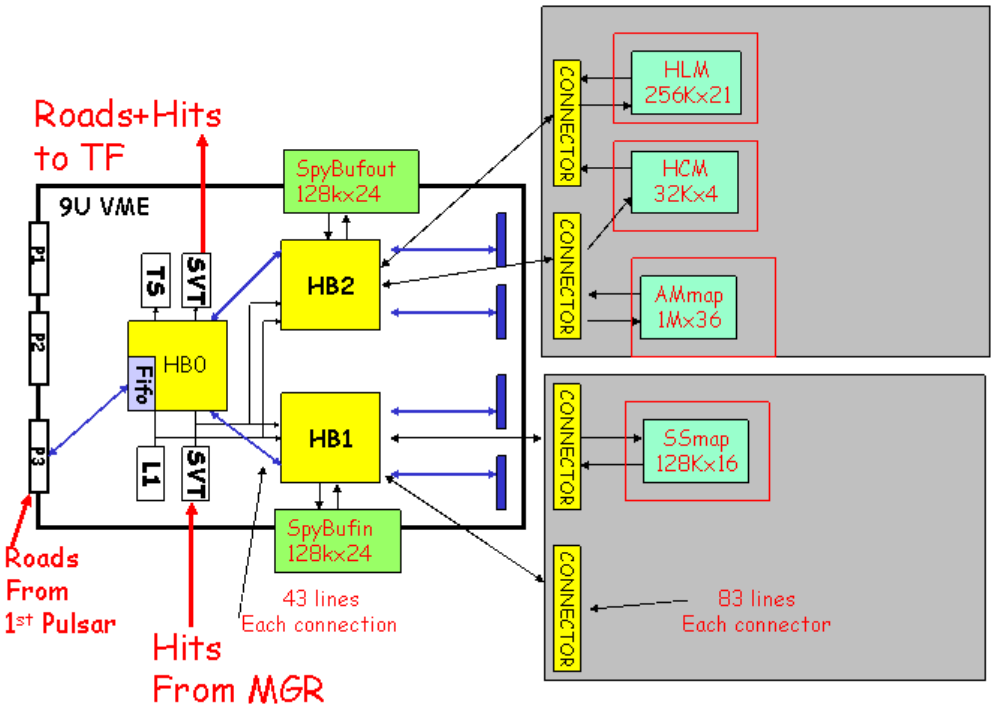

Figure 14: Hit Buffer on a Pulsar board.

Roads produced by the AMS-RW are received from a transition board and stored in a small input FIFO (512 locations) implemented in the Control FPGA block RAMs. A small input Spy Buffer for roads is also located inside the Pulsar Control FPGA.

Hits sent by the Merger board are received from the Pulsar SVT input and stored in a FIFO. A regular SVT Spy Buffer (SpyBufin in figure 14) can be located in the memory directly connected to HB1, one of the two Pulsar data I/O FPGAs.

Road packets are sent to the Pulsar SVT output to be sent to the TF and stored for monitoring in an SVT regular Spy Buffer (SpyBufout in the figure). The Spy Buffer is organized in the memory directly connected to the HB2 data I/O FPGA.

Most of the functionality[5] in the existing Hit Buffer is the same in the new Hit Buffer. The main differences are the following:

1. A larger AM Map is provided to handle roads from an increased associative memory bank. The Road word is enlarged from 18 to 21 bits to a maximum AM bank of 2M roads (1M roads per AM++). The AM Map has also a much larger output data field. While in the old HB single superstrips for different layers are addressed one after the other, in the new AM Map the output data field is large enough (36 bits) to allow 3 superstrips of 12 bits each to be read out in parallel. In conclusion, the AM Map has two locations for each road, to serve the 6 layer configuration (4M x 36). For 512K patterns, a 1M x 36 AM Map memory would be used.
2. We simplified the logic that builds the address for the Hit List Memory. The HB has the capability of storing more than one hit for each superstrip (one Hit List per superstrip). The maximum number of words in the Hit List is fixed at 8. The Hit Count keeps track of the number of words currently stored in each Hit List acting as a set of counters. The Hit Count was implemented as a Tag Ram using now obsolete chips. This allowed a fast reset of all counters at the end of each event. Each counter (one per superstrip) was implemented as a memory location where the current number of hits belonging to a certain superstrip was saved. At the arrival of a new hit, the related current number of hits had to be read, incremented and written back.

   The new AM++ avoids the read-modify-write cycle and makes it possible to build a HB without Tag Ram that is now unavailable. The AM++ tells the HB which of the six layers have hits (bitmap, see AMS-RW section). This means that the HB will know whether a superstrip is empty or not. The Hit Counter logic can be implemented as real counters.

   The HB first counts the hits in each superstrip and then writes the superstrip counter content in the appropriate HCM location. Hits arrive in random order. There are 31 streams (one for each physical module: 6 electrical barrels times 5 layers plus XFT), but they are merged by the Merger. Each stream is sorted, so hits belonging to the same superstrip are consecutive. The HB in order to do the counting of hits per superstrip must have 31 counters, one for each stream, so that those belonging to the same superstrip will arrive one after the other. The arrival of a new hit will cause the increment of the appropriate counter and an update of the corresponding HCM location. The arrival of a hit belonging to a new superstrip will signal the end of the previous list, and the corresponding

counter will be reset to start to work on the new superstrip. The change of superstrip will be detected by comparing the output of the SS Map and ML ADD shown in figure 13. The Hit Count Memory does not need to be reset at the end of the event since the HB will be able to overwrite the Hit Count Memory content of an old event with the up-to-date content of a new event. Only the locations corresponding to empty superstrips will not be overwritten. The HB, however, will never access them since it knows which superstrips are empty (bitmap provided by the AMS).

3. Faster clock. The 3 Pulsar FPGAs provide large space for HB logic and a lot of flexibility. Pipeline stages can be added to the minimal architecture structure shown in figure 13 to constrain the clock cycle as we require. The speed limit of the new SVT will likely be the I/O between the HB and TF. The SVT I/O speed limit is ~40 MHz.
4. Flexibility for error handling. The integration of the logic in 3 large highly interconnected FPGAs gives flexibility for future evolution of the system.

4.4 Track Fitter

Once a road is found with a sufficient numbers of hits, the track candidate has to be fit. Because the road is thin and thus has a narrow range of track angles and curvature, track fitting can be done with a linear approximation. There are six measured hit coordinates (hit positions in four silicon layers plus the curvature and azimuth of the drift chamber track), and six track parameters to be determined (impact parameter relative to the beam line: *d*, curvature: *c*, azimuth: ϕ, and 3 constraints which can be combined to form a track fit χ^2: χ_1, χ_2, and χ_3). The relationship between them is expressed in terms of six linear equations:

$$P_j = P_j(\vec{x}) = \vec{F}_j \cdot \vec{x} + Q_j \qquad (1)$$

where $\vec{x}$ is the vector of the 6 hit coordinates and P_j is the vector of track parameters. $\vec{F}_j$ and Q_j are constants that depend only on the detector geometry and the magnetic field.

For a given road, equation 1 can be rewritten to simplify the calculation in the Track Fitter. Let the track coordinate x be the sum of the coordinate at the edge of the superstrip for that road, x_0, and the position of the hit relative to the superstrip edge, δx. Then equation 1 becomes

$$P_j = P_{0j} + \delta P_j = \vec{F}_j \cdot (\vec{x}_0 + \delta\vec{x}) + Q_j \qquad (2)$$

where $P_{0j} = \vec{F}_j \cdot \vec{x}_0 + Q_j$ is a constant for that road and $\delta P_j = \vec{F}_j \cdot \delta\vec{x}$ is to be calculated for each track. The P_{0j} ("intercepts") and $\vec{F}_j$ ("coefficients") are calculated in advance and stored in memory on the Track Fitter. A track is fit by multiplying the coordinate offsets from the superstrip edge by the coefficients and then adding the intercept.

A functional diagram of the data flow in the Track Fitter is shown in figure 15. Each functional block carries out the following operations:

- **Front End Processor (FEP) -** The data arriving from the Hit Buffer are stored in an Input FIFO. The function of the FEP is to read the SVT data from the Input

FIFO and form an internal Track Fitter word. This word contains all six hit coordinates. In addition, when there are multiple hits in a layer, the FEP creates TF words for all possible hit combinations.

- **TF_FIFO** – The TF_FIFO is a buffer to store the data between the FEP and the downstream fitting processor. With a buffer between the FEP and the Fitter, the Fitter can run asynchronously from the FEP.
- **Fitter** – The function of the Fitter is to calculate the track parameters. The Fitter contains a fit controller as well as the hardware to carry out the calculations of the 6 fit parameters and the combined track χ^2.
- **OutPut Processor (OPP)** – The function of the OPP is to send the fit results to the next SVT stage. The OPP collects the necessary information from the TF_FIFO and the Fitter, and packs it into the 7 SVT words that correspond to a track packet. In addition, the OPP makes a track χ^2 cut and takes care of error handling.

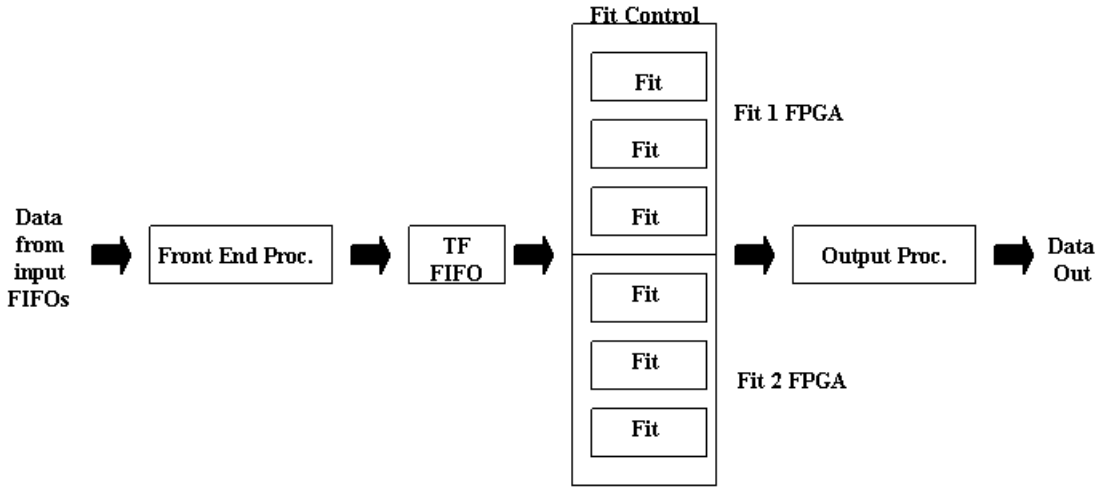

Figure 15: Functional diagram of the Track Fitter data flow.

4.4.1 Track Fitter on a Pulsar board

The existing Track Fitter boards cannot handle a large increase in the number of roads. By moving the functionality to a Pulsar board, we have that capability and can increase the speed of track fitting because of the higher clock speed of the Pulsar. The only hardware task is to design a mezzanine card that can hold the intercept memory. This will be a similar design for all SVT Pulsar boards, with only the requisite memory chips mounted for each application. The firmware in the existing Track Fitters will largely be transferred to the FPGAs on the Pulsar; this work has already started. A sketch of the Track Fitter on a Pulsar board is shown in figure 16.

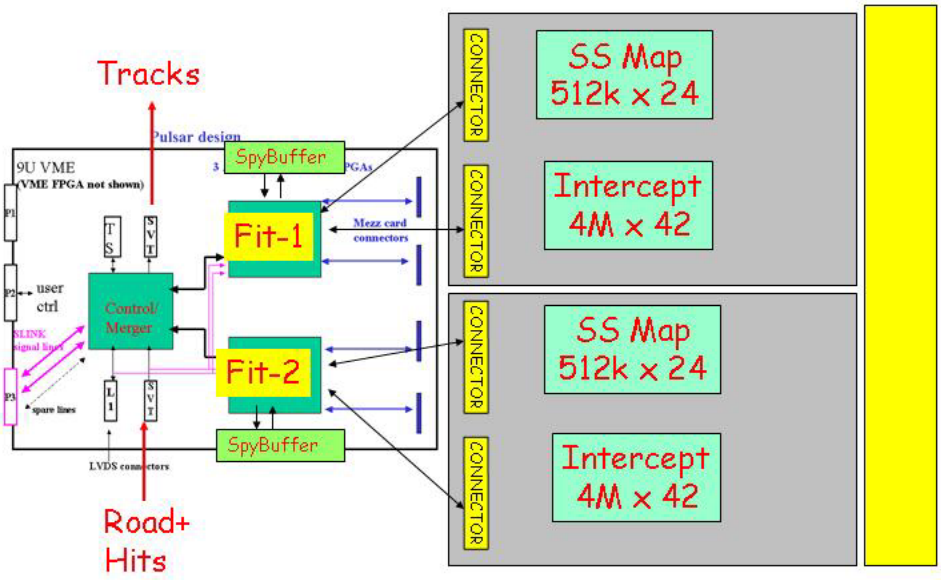

Figure 16: The Track Fitter on a Pulsar board.

The data (roads and hit coordinates) enter the board from the SVT input connector on the front panel. The Pulsar Control FPGA receives the data and stores it in a local memory FIFO. The input spy buffer will be in the Control FPGA.

After converting the data for a track candidate into the internal TF word, the data is sent to both the Fit-1 and Fit-2 FPGAs. The former calculates d, c, and ϕ, while the latter calculates the three χ constraints and the total χ^2. For each calculation, the coefficients are read from memory in the FPGA and the intercepts from the mezzanine memory. Since all the intercepts for a track have the same road address, the three 14-bit intercepts for each FPGA will be fetched in one memory read. There are more than enough mezzanine connector pins for the address and data lines. The superstrip edges, which are subtracted from the hit coordinates, are stored in the SSMap memory on the mezzanine. The results of the calculations are stored in two output spy buffers, one on each of the fit FPGAs.

The results of the fitting calculations are sent from the Fit FPGAs back to the Control FPGA. There the data is prepared to exit the Track Fitter board. A small local spy buffer will be used to augment the deeper spy buffers on each Fit FPGA. The data leave the board through the output SVT connector on the Pulsar front panel.

4.5 Software

The existing SVT software consists of simulation, online code (*e.g.* downloading constants), diagnostic tools for the online and offline environments, generation of patterns and constants, and database access code. The functionality will not change, but the code will have to be modified for the new hardware. We do not as yet have a list of tasks matched with individuals; that is in progress. However we do have significant resources we can apply to these tasks, namely the Wisconsin group, M. Rescigno, A. Cerri, and S. Donati, with help from those working on the firmware design.

5. Cost Estimate

The following is the cost estimate for the SVT upgrade. Included are the costs to be incurred by the U.S. Not included are the AM++ boards for which INFN funds have been obtained. The total should be reasonably accurate because most of the hardware cost is based on the recent procurement of Pulsar boards, transition cards, and mezzanine cards. Note that we will build 2 Pulsar boards per wedge (plus spares) because one set of Pulsar boards has recently been integrated into the SVT as a first phase Road Warrior.

- Pulsar boards
  (2/wedge x (12 wedges + spare + test stand) = 28) @ $4.1K/ $115K
- Mezzanine cards
  AMS-RW: 14 boards @ $200/
  HB: 14 @ $100/, 14 @ $200/
  TF: 28 @ $750/
  28K
- Transition cards for Hit Buffer
  (1/board x (12+1+1) = 14) @ $250/ 4K
- Cables and connectors 1K
- Engineering (3 months) 35K
- Firmware engineering (8 months) 37K

Total $220K

6. Personnel and Schedule

6.1 AM++

Personnel:

A. Bardi AM++ board and LAMB (mezzanine for AM chips)
L. Tripiccione AM chip
A. Annovi diagnostic and control software
P. Giovacchini, I. Ruffilli (students)
(A. Annovi and P. Giovacchini will be at Fermilab starting in January, 2005. The rest of the group will be at Fermilab starting in June, 2005.)

Schedule:

AM chip:
- Send prototype to fabricator (end of 6/04).
- Prepare chip & LAMB test software; test with FPGAs (6/04-9/04).
- Test with standard-cell chips (10/04-12/04).
- Production in Italy (1/05-5/05)

LAMB:
- Prototype ready for testing (end of 6/04)
- Production in Italy (1/05-5/05)

AM++ board:
- Prototype ready for testing (7/04)

Final prototype testing (10/04-11/04)
Testing prototype at Fermilab including Run Control (1/05-5/05)
Production in Italy (1/05-5/05)
Install and test system at Fermilab (6/05-9/05)

6.2 AM Sequencer/Road Warrior

Personnel:
F. Spinella
M. Piendibene
Schedule:
Sequencer firmware simulated (11/04)
Test with AM++ prototype (12/04)
Road Warrior firmware simulated (2/05)
All firmware complete (4/05)
Test in Pisa (5/05)
Test at Fermilab (6/05-7/05)
Install entire system and test (7/05-9/05)

6.3 Hit Buffer

Personnel:
I. Furic
T. Maruyama
T. Mansikkala (firmware engineer)
Schedule:
Detailed firmware specification (7/04-9/04)
Write firmware (10/04-3/05)
Test in test stand (4/05-5/05)
Test with SVT (6/05-7/05)
Install and test full system (8/05-9/05)

6.4 Track Fitter

Personnel:
J. Adelman
U. Yang
Schedule:
Firmware specification re: changes from existing TF (7/04-8/04)
Write firmware (9/04-3/05)
Test in test stand (4/05-5/05)
Test with SVT (6/05-7/05)
Install and test full system (8/05-9/05)

6.5 Pulsar mezzanine cards

Personnel:
F. Tang (engineer)
Schedule:
Receive agreed upon common specification for I/O, memory (8/04)
Design boards and simulate (8/04-11/04)
Produce and test prototype (10/04-12/04)
Production (11/04-3/05)

6.6 Software

Personnel:
Wisconsin group
M. Rescigno
A. Cerri
S. Donati
help from those working on the firmware design.
Tasks: (a schedule has not yet been prepared; it is in progress)
List of requirements
Simulation (svtsim)
Patterns, constants, and high-level files
Online diagnostics (including spydumps/spymon/svtmon)
Online software (svtvme, cold start)
Readout
Offline basic tools
Infrastructure (including database, configuration files)

7. Conclusion

The execution time of the SVT will limit the physics capability of CDF with run IIb luminosity. To mitigate the problem, we propose upgrading the SVT based on the existing Pulsar board and the AM++ system that was developed for the LHC. We can complete the project on time and within budget because little new hardware has to be built and the new firmware is based on that in the existing SVT system.

[1] For the latest documentation, see http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/svtupgrade/index.html.
[2] See http://www-cdf.fnal.gov/upgrades/daq_trig/trigger/svt/index.html.
[3] See http://hep.uchicago.edu/~thliu/projects/Pulsar/.
[4] See http://www.europractice.imec.be.
[5] See http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/BoardDocs/HitBuffer/specs/.